PRIVACY-PRESERVING PUBLIC AUDITING FOR DATA STORAGE SECURITY IN CLOUD COMPUTING

T.Vidhya Sree

M.Phil Research Scholar ,CS, MGR College, Hosur (India)

ABSTRACT

Cloud Computing is the long dreamed vision of computing as a utility, where users can remotely store their data into the cloud so as to enjoy the on-demand high quality applications and services from a shared pool of configurable computing resources. By data outsourcing, users can be relieved from the burden of local data storage and maintenance. However, the fact that users no longer have physical possession of the possibly large size of outsourced data makes the data integrity protection in Cloud Computing a very challenging and potentially formidable task, especially for users with constrained computing resources and capabilities. Thus, enabling public auditability for cloud data storage security is of critical importance so that users can resort to an external audit party to check the integrity of outsourced data when needed. To securely introduce an effective third party auditor (TPA), the following two fundamental requirements have to be met: 1) TPA should be able to efficiently audit the cloud data storage without demanding the local copy of data, and introduce no additional on-line burden to the cloud user; 2) The third party auditing process should bring in no new vulnerabilities towards user data privacy. In this paper, we utilize the public key based homomorphic authenticator and uniquely integrate it with random mask technique to achieve a privacy-preserving public auditing system for cloud data storage security while keeping all above requirements in mind. To support efficient handling of multiple auditing tasks, we further explore the technique of bilinear aggregate signature to extend our main result into a multi-user setting, where TPA can perform multiple auditing tasks simultaneously. Extensive security and performance analysis shows the proposed schemes are provably secure and highly efficient.

I INTRODUCTION

Cloud Computing has been envisioned as the next-generation architecture of IT enterprise, due to its long list of unprecedented advantages in the IT history: on-demand self-service, ubiquitous network access, location independent resource pooling, rapid resource elasticity, usage-based pricing and transference of risk [1]. As a disruptive technology with profound implications, Cloud Computing is transforming the very nature of how businesses use information technology. One fundamental aspect of this paradigm shifting is that data is being centralized or outsourced into the Cloud. From users' perspective, including both individuals and IT enterprises, storing data remotely into the cloud in a flexible on-demand manner brings appealing benefits: relief of the burden for storage management, universal data access with independent geographical locations, and avoidance of capital expenditure on hardware, software, and personnel maintenances, etc [2].

While these advantages of using clouds are unarguable, due to the opaqueness of the Cloud—as separate administrative entities, the internal operation details of cloud service providers (CSP) may not be known by cloud users—data outsourcing is also relinquishing user's ultimate control over the fate of their data. As a result, the correctness of the data in the cloud is being put at risk due to the following reasons. First of all, although the infrastructures under the cloud are much more powerful and reliable than personal computing devices, they are still facing the broad range of both internal and external threats for data integrity. Examples of outages and security breaches of noteworthy cloud services appear from time to time [3–4]. Secondly, for the benefits of their own, there do exist various motivations for cloud service providers to behave unfaithful towards the cloud users regarding the status of their outsourced data. Examples include cloud service providers, for monetary reasons, reclaiming storage by discarding data that has not been or is rarely accessed, or even hiding data loss incidents so as to maintain a reputation [7–9]. In short, although outsourcing data into the cloud is economically attractive for the cost and complexity of long-term large-scale data storage, it does not offer any guarantee on data integrity and availability.

This problem, if not properly addressed, may impede the successful deployment of the cloud architecture.

Therefore, how to enable a privacy-preserving third-party auditing protocol, independent to data encryption, is the problem we are going to tackle in this paper. Our work is among the first few ones to support privacy-preserving public auditing in Cloud Computing, with a focus on data storage. Besides, with the prevalence of Cloud Computing, a foreseeable increase of auditing tasks from different users may be delegated to TPA. As the individual auditing of these growing tasks can be tedious and cumbersome, a natural demand is then how to enable TPA to efficiently perform the multiple auditing tasks in a batch manner, i.e., simultaneously. To address these problems, our work utilizes the technique of public key based homomorphic authenticator [5, 6, 7], which enables TPA to perform the auditing without demanding the local copy of data and thus reduces the communication and computation overhead as compared to the straightforward data auditing approaches. By integrating the homomorphic authenticator with random mask technique, our protocol guarantees that TPA could not learn any knowledge about the data content stored in the cloud server during the efficient auditing process. The aggregation and algebraic properties of the authenticator further benefit our design for the batch auditing. Specifically, our contribution in this work can be summarized as the following three aspects:

1) We motivate the public auditing system of data storage security in Cloud Computing and provide a privacy-preserving auditing protocol, i.e., our scheme supports an external auditor to audit user's outsourced data in the

cloud without learning knowledge on the data content.

2) To the best of our knowledge, our scheme is the first to support scalable and efficient public auditing in the Cloud Computing. In particular, our scheme achieves batch auditing where multiple delegated auditing tasks from different users can be performed simultaneously by the TPA.

3) We prove the security and justify the performance of our proposed schemes through concrete experiments and comparisons with the state-of-the-art.

II PROBLEM STATEMENT

The cloud user , who has large amount of data files to be stored in the cloud; the cloud server (CS), which is managed by cloud service provider (CSP) to provide data storage service and has significant storage space and computation resources (we will not differentiate CS and CSP hereafter.); the third party auditor (TPA), who has expertise and capabilities that cloud users do not have and is trusted to assess the cloud storage service security on behalf of the user upon request. Users rely on the CS for cloud data storage and maintenance. They may also dynamically interact with the CS to access and update their stored data for various application purposes. The users may resort to TPA for ensuring the storage security of their outsourced data, while hoping to keep their data private from TPA. We consider the existence of a semi-trusted CS in the sense that in most of time it behaves properly and does not deviate from the prescribed protocol execution. While providing the cloud data storage based services, for their own benefits the CS might neglect to keep or deliberately delete rarely accessed data files which belong to ordinary cloud users. Moreover, the CS may decide to hide the data corruptions caused by server hacks or Byzantine failures to maintain reputation. We assume the TPA, who is in the business of auditing, is reliable and independent, and thus has no incentive to collude with either the CS or the users during the auditing process. TPA should be able to efficiently audit the cloud data storage without local copy of data and without bringing in additional on-line burden to cloud users. However, any possible leakage of user's outsourced data towards TPA through the auditing protocol should be prohibited.

III. DESIGN GOALS

To enable privacy-preserving public auditing for cloud data storage under the aforementioned model, our protocol design should achieve the following security and performance guarantee:

1) Public auditability: to allow TPA to verify the correctness of the cloud data on demand without retrieving a copy of the whole data or introducing additional on-line burden to the cloud users;

2) Storage correctness: to ensure that there exists no cheating cloud server that can pass the audit from TPA without indeed storing users' data intact;

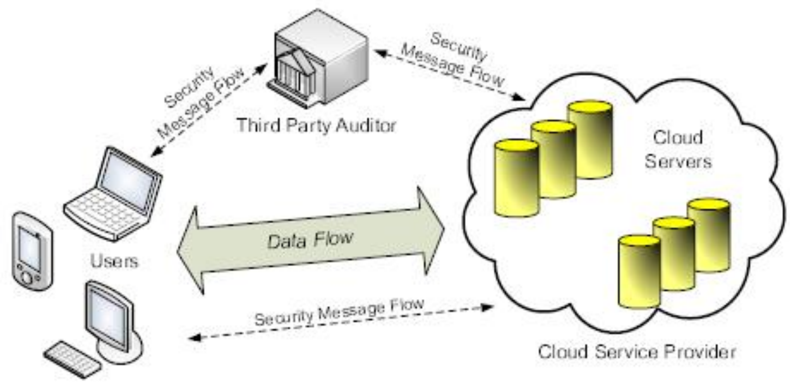

Fig 3 The architecture of cloud data storage service

3) Privacy-preserving: to ensure that there exists no way for TPA to derive users' data content from the information collected during the auditing process;

4) Batch auditing: to enable TPA with secure and efficient auditing capability to cope with multiple auditing delegations from possibly large number of different users simul-taneously;

5) Lightweight: to allow TPA to perform auditing with minimum communication and computation overhead.

3.1 The Proposed Schemes

In the introduction we motivated the public auditability with achieving economies of scale for cloud computing. This section presents our public auditing scheme for cloud data storage security. We start from the overview of our public auditing system and discuss two straightforward schemes and their demerits. Then we present our main result for privacy-preserving public auditing to achieve the aforementioned design goals. We also show how to extent our main scheme to support batch auditing for TPA upon delegations from multi-users. Finally, we discuss how to adapt our main result to support data dynamics.

3.1.1 Definitions and Framework of Public Auditing System

We follow the similar definition of previously proposed schemes in the context of remote data integrity checking [5,7,8] and adapt the framework for our privacy-preserving public auditing system.5

A public auditing scheme consists of four algorithms (KeyGen, SigGen, GenProof, VerifyProof). KeyGen is a key generation algorithm that is run by the user to setup the scheme. SigGen is used by the user to generate verification metadata, which may consist of MAC, signatures, or other related information that will be used for auditing. GenProof is run by the cloud server to generate a proof of data storage correctness, while VerifyProof is run by the TPA to audit the proof from the cloud server.

Our public auditing system can be constructed from the above auditing scheme in two phases, Setup and Audit:

- Setup: The user initializes the public and secret parameters of the system by executing KeyGen, and pre-processes the data file F by using SigGen to generate the verification metadata. The user then stores the data file F at the cloud server, deletes its local copy, and publishes the verification metadata to TPA for later audit. As part of pre-processing, the user may alter the data file F by expanding it or including additional metadata to be stored at server.
- Audit: The TPA issues an audit message or challenge to the cloud server to make sure that the cloud server has retained the data file F properly at the time of the audit. The cloud server will derive a response message from a function of the stored data file F by executing GenProof. Using the verification metadata, the TPA verifies the response via VerifyProof.

Note that in our design, we do not assume any additional property on the data file, and thus regard error-correcting codes as orthogonal to our system. If the user wants to have more error-resiliency, he/she can first redundantly encode the data file and then provide us with the data file that has error-correcting codes integrated.

3.1.2 The Basic Schemes

Before giving our main result, we first start with two warmup schemes. The first one does not ensure privacy-preserving guarantee and is not as lightweight as we would like. The second one overcomes the first one, but suﬀers from other undesirable systematic demerits for public auditing: bounded usage and auditor statefulness, which may pose additional on-line burden to users as will be elaborated shortly. We believe the analysis of these basic schemes will lead us to our main result, which overcomes all these drawbacks.

During the Audit phase, the TPA requests from the cloud server a number of randomly selected blocks and their corresponding MACs to verify the correctness of the data file. The insight behind this approach is that auditing most of the file is much easier than the whole of it. However, this simple solution suﬀers from the following severe drawbacks:

1) The audit from TPA demands retrieval of users' data, which should be prohibitive because it violates the privacy-preserving guarantee;

2) Its communication and computation complexity are both linear with respect to the sampled data size, which may result in large communication overhead and time delay, especially when the bandwidth available between the TPA and the cloud server is limited.

The cloud server and ask for a fresh keyed MAC for comparison, thus achieving privacy-preserving auditing. However, in this method: 1) the number of times a particular data file can be audited is limited by the number of secret keys that must be a fixed priori. Once all possible secret keys are exhausted, cloud user then has to retrieve data from the server in order to re-compute and re-publish new MACs to TPA. 2) The TPA has to maintain and update state between audits, i.e., keep a track on the possessed MAC keys. Considering the potentially large number of audit delegations from multiple users, maintaining such states for TPA can be difficult and error prone.

3.1.3 The Privacy-Preserving Public Auditing Scheme

To effectively support public auditability without having to retrieve the data blocks themselves, we resort to the homomorphic authenticator technique [5, 6, 7]. Homomorphic authenticators are unforgeable verification metadata generated from individual data blocks, which can be securely aggregated in such a way to assure an auditor that a linear combination of data blocks is correctly computed by verifying only the aggregated authenticator. However, the direct adoption of these techniques is not suitable for our purposes, since the linear combination of blocks may potentially reveal user data information, thus violating the privacy-preserving guarantee. Specifically, if enough number of the linear combinations of the same blocks are collected, the TPA can simply derive the user's data content by solving a system of linear equations.

Overview To achieve privacy-preserving public auditing, we propose to uniquely integrate the homomorphic authenticator with random mask technique. In our protocol, the linear combination of sampled blocks in the server's response is masked with randomness generated by a pseudo random function (PRF). With random mask, the TPA no longer has all the necessary information to build up a correct group of linear equations and therefore cannot derive the user's data content, no matter how many linear combinations of the same set of file blocks can be collected. Meanwhile, due to the algebraic property of the homomorphic authenticator, the correctness validation of the block-authenticator pairs will not be affected by the randomness generated from a PRF, which will be shown shortly. Note that in our design, we use public key based homomorphic authenticator, specifically, the one in [7] which is based on BLS signature [9], to equip the auditing protocol with public auditability. Its flexibility in signature aggregation will further benefit us for the multi-task auditing.

3.1.4 Implementation

We motivate the public auditing system of data storage security in Cloud Computing and provide a privacy-preserving auditing protocol. Our scheme enables an external auditor to audit user's cloud data without learning the data content.

To the best of our knowledge, our scheme is the first to support scalable and efficient privacy- preserving public storage auditing in Cloud. Specifically, our scheme achieves batch auditing where multiple delegated auditing tasks from different users can be performed simultaneously by the TPA in a privacy-preserving manner.

One of the important concerns that need to be addressed is to assure the customer of the integrity i.e. correctness of his data in the cloud. As the data is physically not accessible to the user the cloud should provide a way for the user to check if the integrity of his data is maintained or is compromised.

In this paper we provide a scheme which gives a proof of data integrity in the cloud which the customer can employ to check the correctness of his data in the cloud. This proof can be agreed upon by both the cloud and the customer and can be incorporated in the Service level agreement (SLA).

It is important to note that our proof of data integrity protocol just checks the integrity of data i.e. if the data has been illegally modified or deleted.

3.1.5 Key Features

- Apart from reduction in storage costs data outsourcing to the cloud also helps in reducing the maintenance.
- Avoiding local storage of data.
- By reducing the costs of storage, maintenance and personnel.
- It reduces the chance of losing data by hardware failures.
- Unauthorized person unable to fetch the data without owner knowledge.
- High data reliability and Dynamic security provided.
- Data retrieval from the cloud is highly securable.

IV CONCLUSION

In this paper, we propose a privacy-preserving public auditing system for data storage security in Cloud Computing, where TPA can perform the storage auditing without demanding the local copy of data. We utilize the homomorphic authenticator and random mask technique to guarantee that TPA would not learn any knowledge about the data content stored on the cloud server during the efficient auditing process, which not only eliminates the burden of cloud user from the tedious and possibly expensive auditing task, but also alleviates the users' fear of their outsourced data leakage. Considering TPA may concurrently handle multiple audit sessions from different users for their outsourced data files, we further extend our privacy-preserving public auditing protocol into a multi-user setting, where TPA can perform the multiple auditing tasks in a batch manner, i.e., simultaneously. Extensive security and performance analysis shows that the proposed schemes are provably secure and highly efficient. We believe all these advantages of the proposed schemes will shed light on economies of scale for Cloud Computing.

REFERENCES

1. P. Mell and T. Grance, "Draft NIST working definition of cloud computing," Referenced on June. 3rd, 2009 Online at http://csrc.nist.gov/groups/SNS/cloud-computing/index.html, 2009.
2. M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the clouds: A berkeley view of cloud computing," University of California, Berkeley, Tech. Rep. UCB-EECS-2009-28, Feb 2009.
3. N. Gohring, "Amazon's s3 down for several hours," Online at http://www.pcworld.com/ businesscenter/article/142549/amazons s3 down for several hours.html, 2008.
4. B. Krebs, "Payment Processor Breach May Be Largest Ever," Online at http://voices.washingtonpost.

com/securityfix/2009/01/payment processor breach may b.html, Jan. 2009.

5. G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," Cryptology ePrint Archive, Report 2007/202, 2007, http://eprint. iacr.org/.
6. Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling public verifiability and data dynamics for storage security in cloud computing," in **Proc. of ESORICS'09**, Saint Malo, France, Sep. 2009.
7. H. Shacham and B. Waters, "Compact proofs of retrievability," in **Proc. of Asiacrypt 2008**, vol. 5350, Dec 2008, pp. 90–107
8. A. Juels and J. Burton S. Kaliski, "Pors: Proofs of retrievability for large files," in **Proc. of CCS'07**, Alexandria, VA, October 2007, pp. 584–597.
9. D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the weil pairing," in Proc. of ASI- ACRYPT'01. London, UK: Springer-Verlag, 2001, pp. 514–532.